Artificial Intelligence

k - Nearest Neighbors

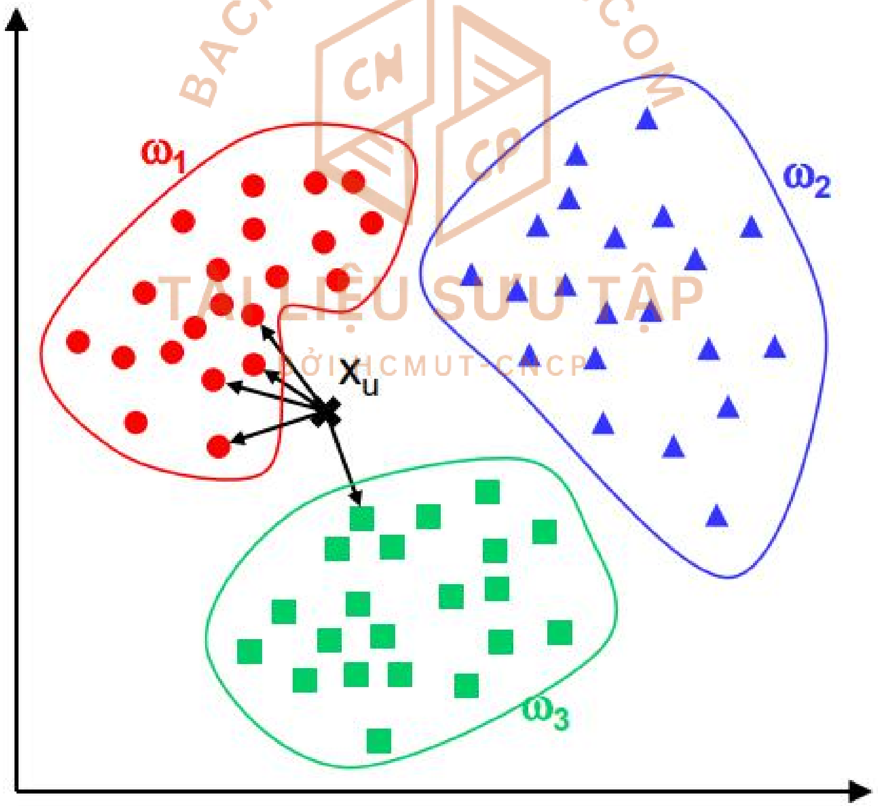
Pham Viet Cuong
Dept. Control Engineering & Automation, FEEE
Ho Chi Minh City University of Technology

k - Nearest Neighbors

- A type of supervised ML algorithm
- Can be used for both classification and regression
- Lazy learning algorithm

k - Nearest Neighbors

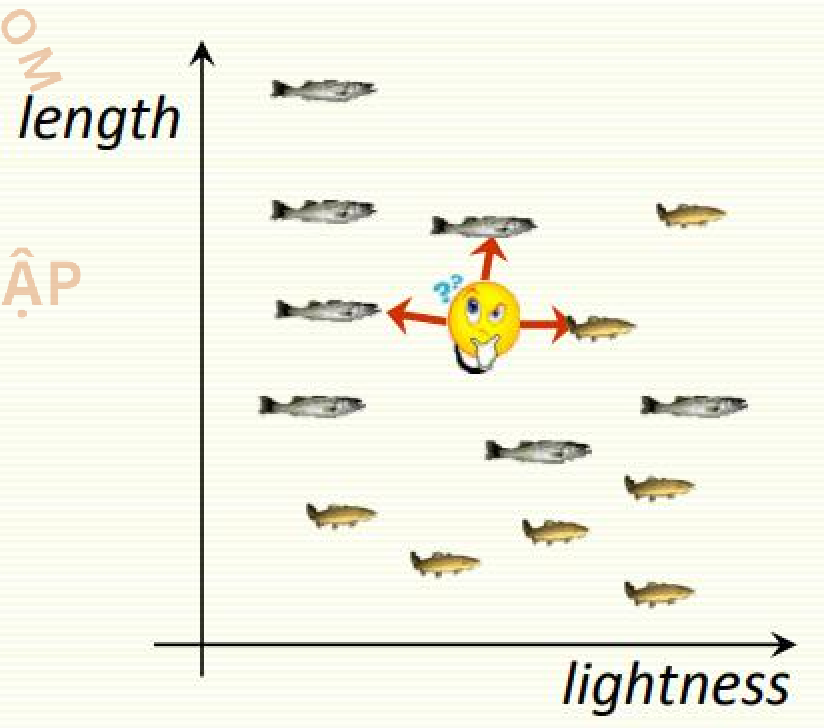
- ✓ Uses 'feature similarity' to predict the values of new datapoints
- ✓ The new data point will be assigned a value based on how closely it matches the points in the training set



k - Nearest Neighbors

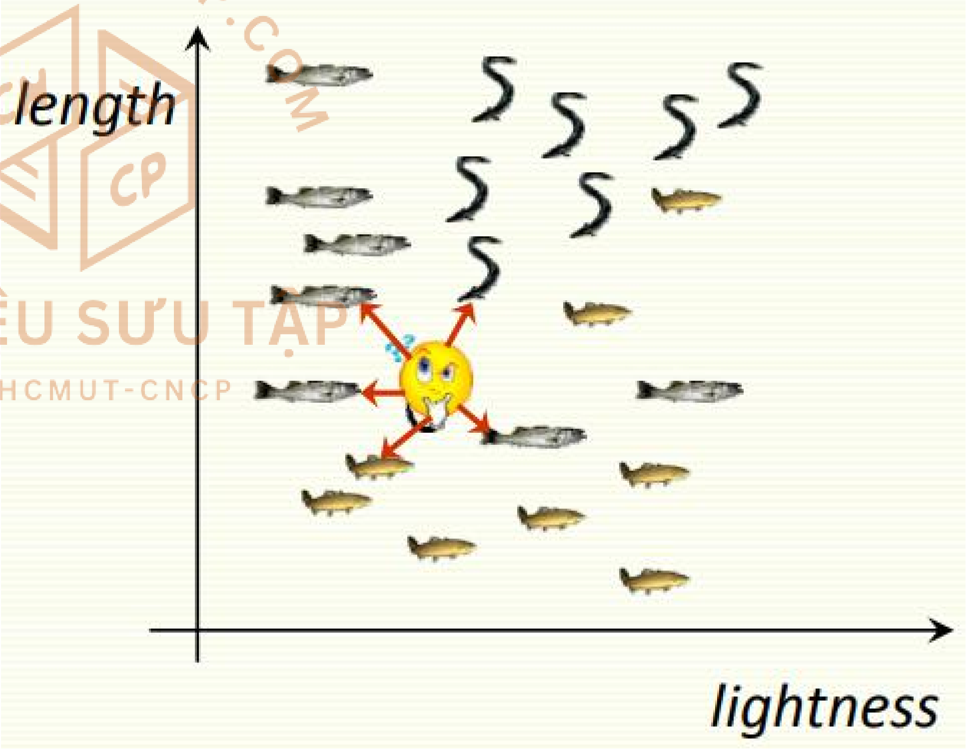
- The kNN requires
  - An integer k
  - A set of labeled examples (training data)
  - A metric to measure “closeness”

- Example 1: Classification
  - 2D
  - 2 classes
  - k = 3
  - Euclidean distance
  - 2 sea bass, 1 salmon



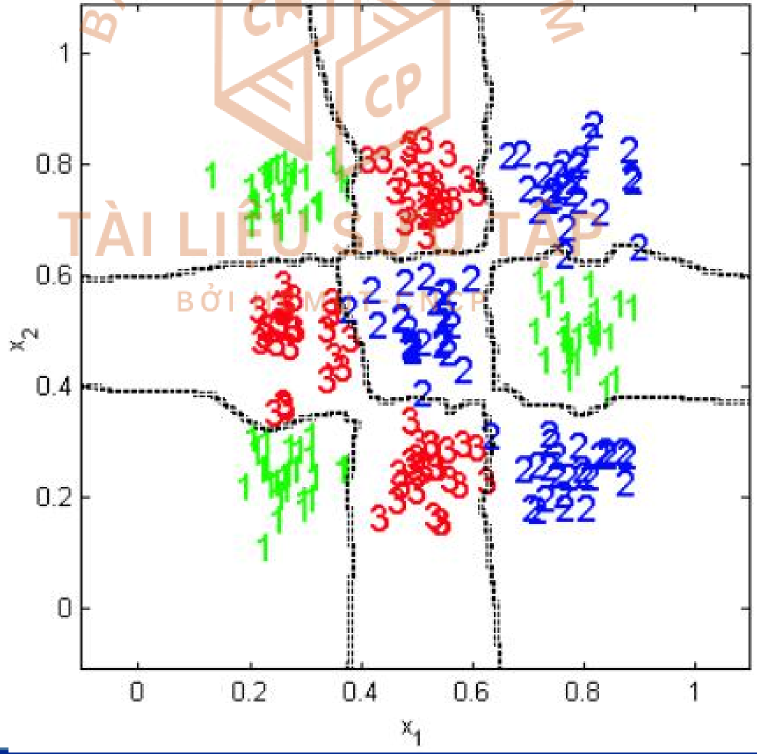
k - Nearest Neighbors

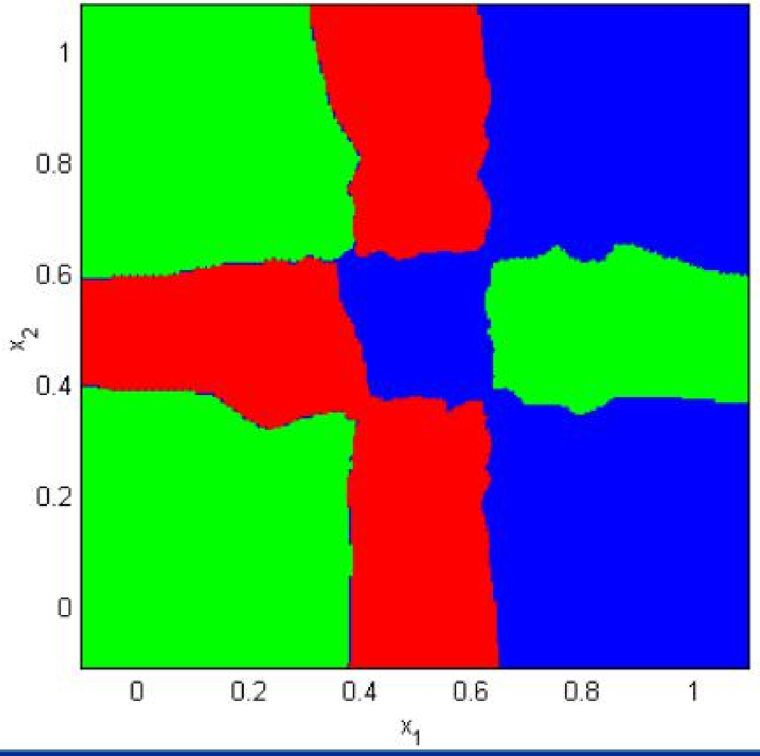
- Example 2: Classification
  - 2D
  - Three classes
  - k = 5
  - Euclidean distance



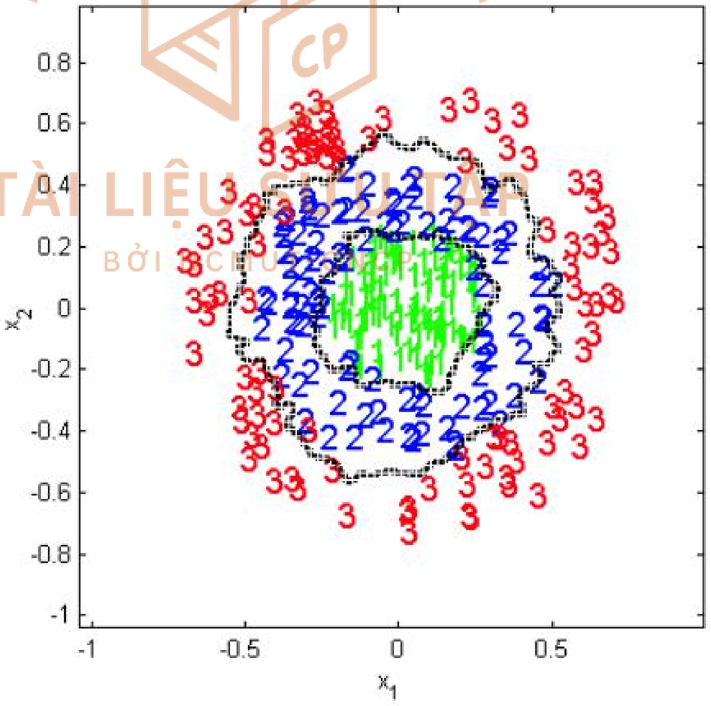
k - Nearest Neighbors

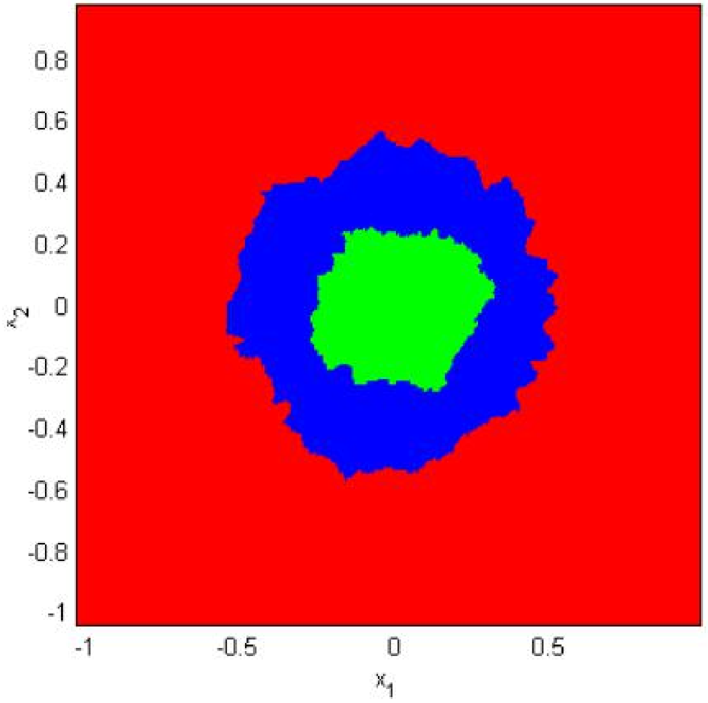
✓ Example 3: Classification
- Three-class 2D problem
- non-linearly separable
- k = 5
- Euclidean distance





- Example 4: Classification
  - Three-class 2D problem
  - non-linearly separable
  - k = 5
  - Euclidean distance





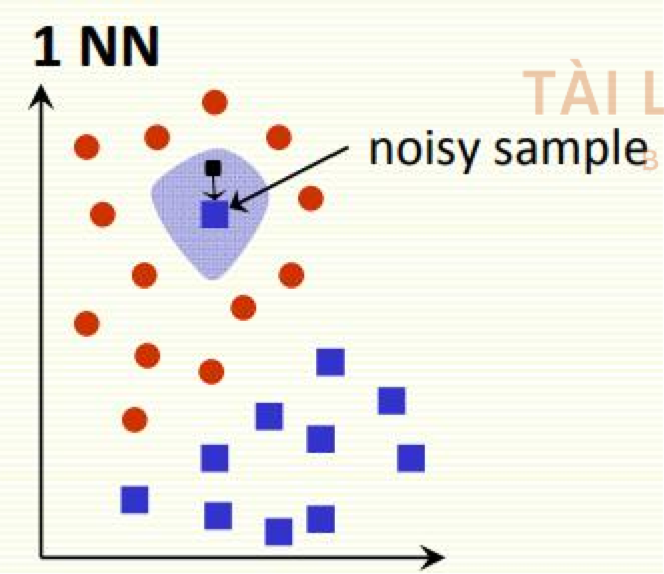
k - Nearest Neighbors

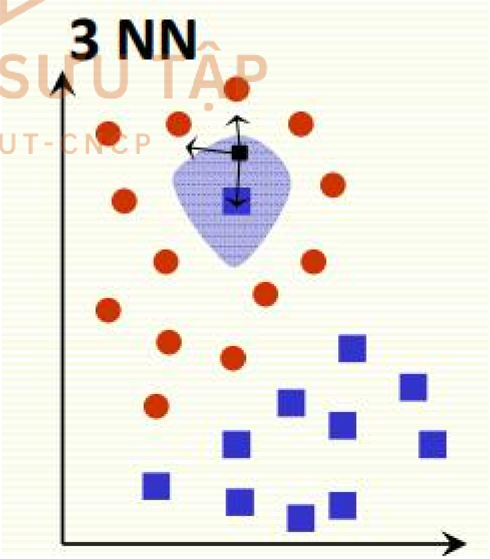
- Algorithm
  - Step 1: Load training data and test data
  - Step 2: Choose k
  - Step 3:
    - Calculate distance between test data and other data points
    - Identify k nearest neighbors
    - Use class labels of nearest neighbors to determine the class label of test data (e.g., by taking majority vote)
  - Step 4: End

k - Nearest Neighbors

✓ How to choose k?

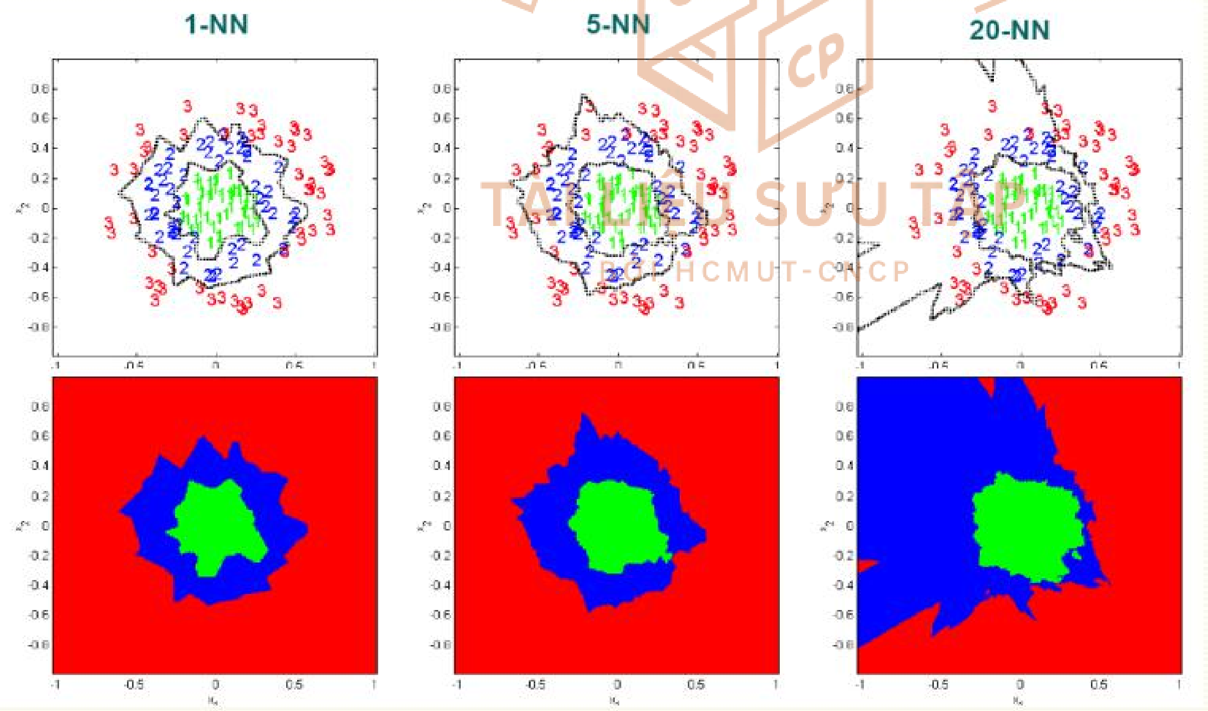
- If infinite number of samples available, the larger is k the better
- In practice: # samples is finite
- Rule of thumb: k = sqrt(n), n: number of examples
- k = 1: for efficiency, but can be sensitive to “noise”





k - Nearest Neighbors

- How to choose k?
  - Larger k may improve performance, but too large k destroys locality
  - Smaller k: higher variance (less stable)
  - Larger k: higher bias (less precise)



k - Nearest Neighbors

- How well does KNN work?
  - If we have lots of samples, kNN works well

k - Nearest Neighbors

- Minkowski distance

$$D_{Mink}(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

- Mahattan distance

$$MD(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Euclidean distance

$$ED(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

- Chebyshev distance

$$CD(x,y) = \max_{i} |x_i - y_i|$$

k - Nearest Neighbors

✓ Best distance?

Reference	#distances	#datasets	Best distance
[13]	11	8	Manhattan, Minkowski Chebychev Euclidean, Mahalanobis Standardized Euclidean
[62]	3	1	Manhattan
[39]	4	37	Chi square
[72]	18	8	Manhattan, Euclidean, Soergel Contracted Jaccard–Tanimoto Lance–Williams
[52]	5	15	Euclidean and Manhattan
[3]	3	28	Hassanat
[51]	3	2	Hassanat
Ours	54	28	Hassanat

k - Nearest Neighbors

- Euclidian distance

$$ED(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

- Euclidean distance treats each feature as equally important
- However some features (dimensions) may be much more discriminative than others

k - Nearest Neighbors

✓ Euclidian distance

- feature 1 gives the correct class: 1 or 2
- feature 2 gives irrelevant number from 100 to 200
- dataset: **[1 150]**

  [2 110]
- classify **[1 100]**

$$D\left(\begin{bmatrix}1\\100\end{bmatrix},\begin{bmatrix}1\\150\end{bmatrix}\right)=\sqrt{(1-1)^2+(100-150)^2}=50$$

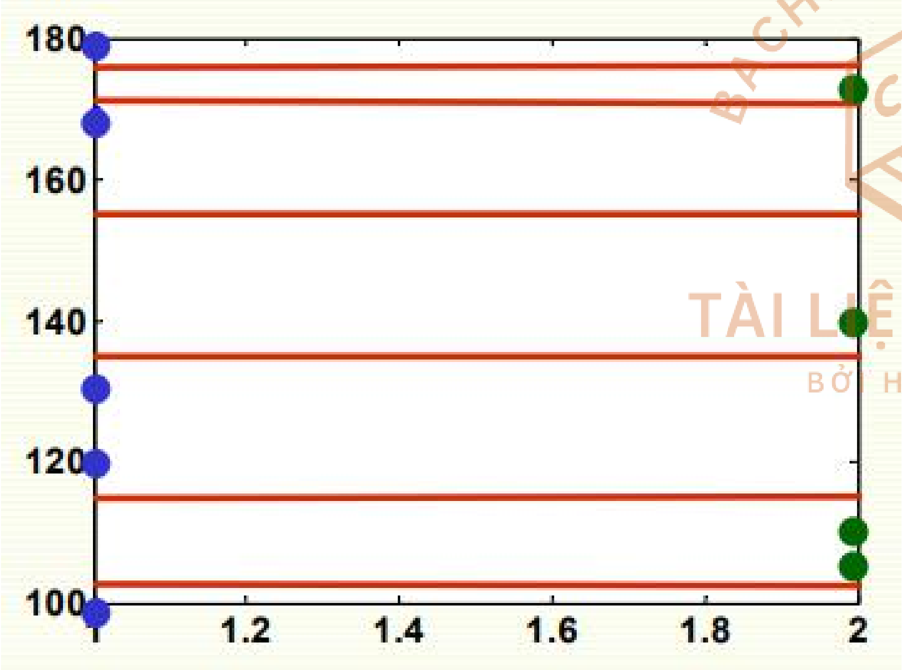
$$D\left(\begin{bmatrix}1\\100\end{bmatrix},\begin{bmatrix}2\\110\end{bmatrix}\right)=\sqrt{(1-2)^2+(100-110)^2}=10.5$$

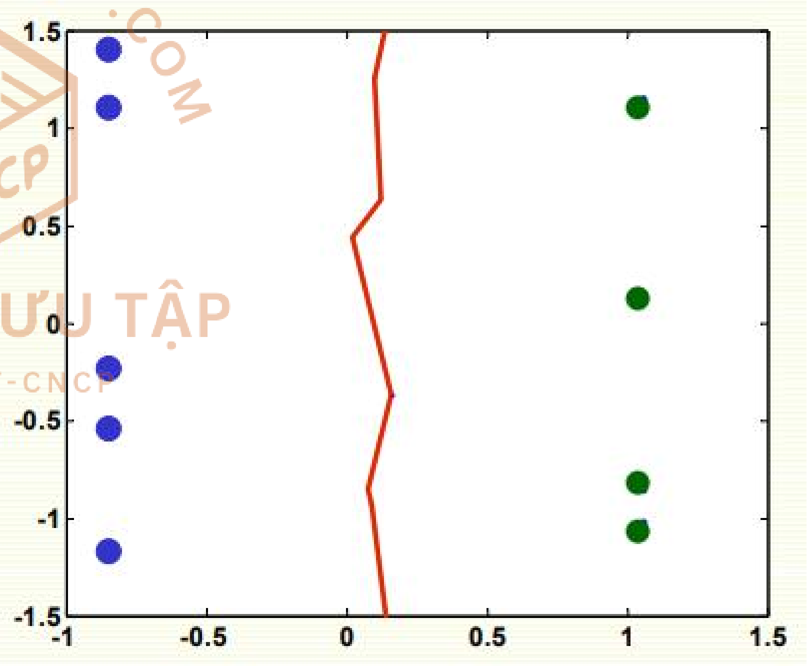
- **[1 100]** is misclassified!
- The denser the samples, the less of this problem
- But we rarely have samples dense enough

k - Nearest Neighbors

✓ Feature nomalization

❖ Linearly scale to 0 mean, variance 1





k - Nearest Neighbors

- Feature weighting
  - Scale each feature by its importance for classification

$$ED(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

w_i

k - Nearest Neighbors

- Computational complexity
  - Basic kNN algorithm stores all examples
  - Very expensive for a large number of samples

k - Nearest Neighbors

- kNN - a lazy learning algorithm
  - Defers data processing until it receives a request to classify unlabeled data
  - Replies to a request for information by combining its stored training data
  - Discards the constructed answer and any intermediate results
  - Lazy algorithms have fewer computational costs than eager algorithms during training but greater storage requirements and higher computational costs on recall

k - Nearest Neighbors

- Advantages
  - Can be applied to the data from any distribution
  - Very simple and intuitive
  - Good classification if the number of samples is large enough
  - Uses local information, which can yield highly adaptive behavior
  - Very easy for parallel implementations
- Disadvantages
  - Choosing k may be tricky
  - Test stage is computationally expensive
  - Need large number of samples for accuracy
  - Large storage requirements
  - Highly susceptible to the curse of dimensionality

k - Nearest Neighbors

- Sources:
  - https://www.csd.uwo.ca/courses/CS4442b/L3-ML-knn.pdf
  - http://research.cs.tamu.edu/prism/lectures/pr/pr_l8.pdf
  - http://web.iitd.ac.in/~bspanda/KNN%20presentation.pdf
  - V. B. Surya Prasath et. al., Effects of Distance Measure Choice on KNN Classifier Performance - A Review, Big Data. 7. 10.1089/big.2018.0175.